REMARKS

The enclosed is responsive to the Examiner's Office Action mailed on April 1, 2009. At the time the Examiner mailed the Office Action claims 126-129, 131-140, 142, and 144-154 were pending. By way of the present response the Applicant has: 1) amended claims 126, 129, 132, 137, and 144-152; 2) added no new claims; and 3) canceled no claims. As such, claims 126-129, 131-140, 142, and 144-154 are now pending. Applicant has amended the claims to clarify the subject matter claimed and respectfully submits that no new matter has been added. For example, support can be found at least on pages 15-16 and 27-28 of the Specification as originally filed. Applicant respectfully requests reconsideration of the present application.

Examiner Interview

Applicant thanks Examiner Basehoar for conducting a telephone interview with Applicant's representatives on April 30, 2009 to discuss proposed amendments to the claims and the significance of the art cited in relation to the claims. In particular, the parties discussed proposed claim amendments highlighting that the form is authored by an independent authoring tool and that the GUI allows identification of one or more actions from a group of two or more actions to be associated with identified input fields. Applicant's representative reiterated that OmniForm is a form authoring tool that alters an existing form rather than automatically generating program code independently of the form authoring tool. Applicant's representative further argued that none of the references describe automatically generating program code that is external to the form. The Examiner agreed to review OminForm and perform a new search. No agreement as to patentability of the claims was made.

Claim Rejections - 35 U.S.C. § 103

Claims 126, 127, 129, 131-134, 136-138, 140, 142, 144-146 and 148-154 stand rejected under 35 U.S.C. 103(a) as being unpatentable over OmniForm User's Manual (hereinafter OmniForm), Caere Corporation, released March 22, 1999 (as evidenced by cited PR NewsWire article), pages 1-108, 173-199, in view of Larsen, U.S. Patent

6,088,700 (hereinafter "Larsen") in further view of Strong et al., U.S. Patent 6,167,523 (hereinafter "Strong").

OmniForm describes the generation of a form and the editing of an existing form by a user. (OmniForm, e.g., pages 25-43.) This editing may be performed through a GUI. (OmniForm, e.g., page 32). In particular, OmniForm describes that fill text objects can be configured with a validation option that allows the form author to select whether a field must be filled in, a type of data to be inputted, etc. (OmniForm, page 76).

Larsen describes a system for automated form completion- i.e., automatically filling in a form with user data to avoid a user needing to reenter the information into multiple forms. The system completes forms by retrieving tagged information previously entered by a user and stored in a database and inserting the information in similarly tagged fields in the form.

Strong describes a computer program for validating input data from an electronic form. In particular, Strong discusses the use of a CGI program, which is external to the form, to perform the validation of user data in a submission of the form and the configuration of such validation in the Registry of the server to which the form is submitted.

Regarding claims 126, 132, and 137, the combination of OmniForm, Larsen, and Strong fails to disclose parsing the received form, providing a graphical user interface to allow identification of one or more actions, and automatically generating program code, all independently of the form authoring tool.

The Office Action relies primarily on OmniForm. OmniForm describes opening and editing an existing form. OmniForm describes this in the context of editing a form authored by OmniForm software (i.e., the form authoring tool). In contrast, the claims describe performing the above listed actions independently of the form authoring tool. OmniForm does describe the ability to convert a scanned paper form, image file, or form created by a Windows application into an OmniForm format (see, e.g., OmniForm, pages 26-32 and 36). OmniForm, however, has to recreate the form in its own format. In contrast, claims 126, 132, and 137 describe parsing **the received form** – i.e., without converting the form, e.g., from a paper format or other electronic format into a different format. Applicant respectfully submits that, in performing the conversion of an existing paper or other electronic form, OmniForm would not be parsing the received form but rather would be converting the form into a format specific to Omniform. Applicant submits that Larsen and Strong also fail to disclose these claim features.

The combination also fails to disclose providing, independently of the form authoring tool, a graphical user interface dependent upon the identified input fields to enable configuration of one or more actions to be carried out in response to a subsequent specific submission of the form, wherein the one or more actions are specified from a group of two or more types of actions. As cited in the Office Action, OmniForm describes setting a validation option for fill text objects. (OmniForm, page 76). In contrast to claims 126, 132, and 137, however, OmniForm performs client-side validation – i.e., validation is performed on the user's computer prior to submitting the form and, therefore, it is not a subsequent specific submission of a specific instance of the form. (see, e.g., ScanSoft OmniForm Internet Suite, page 3, submitted in the IDS dated 1/14/09). Furthermore, client-side validation is not selected from specified from a group of two or more types of actions in OmniForm. Larsen describes automated form completion, but is silent regarding a GUI to allow identification of one or more actions from a group of two or more types of actions. Strong describes a CGI program to perform validation on the submission of a form. Strong, however, is silent regarding a GUI to allow identification of one or more actions from a group of two or more types of actions.

The combination further fails to disclose automatically generating, independently of the form authoring tool, program code to carry out the one or more actions, wherein the program code is external to the form and independent of the form authoring tool. Given that OmniForm is a user's manual lacks detail in regard to the implementation of the described functionality of editing a form, Applicant has provided in an IDS including five archived webpages that further describe the OmniForm 4.0 software. (*see* IDS submitted 1/14/09). "OmniForm automatically creates JavaScript which performs calculations and validation **embedded in the forms in the browser – not at the server**." (ScanSoft OmniForm 4.0: Why should you upgrade?, page 3 and ScanSoft OmniForm 4.0 – Fact Sheet, page 5). The Internet Suite boasts that no CGI is necessary and that client-side validation is supported on the client-side at the browser. (ScanSoft OmniForm Internet Suite, page 3). However, forms "can be processed using 'Get,' which will append the data to the URL specified in the Action option, or using 'Post' to notify the Web server to open the CGI application and pass the data." (ScanSoft OmniForm 4.0: Why should you upgrade?, page 3). While "OmniForm works with CGI scripts on webservers just like regular HTML forms do," **the author or administrator of the form still must provide the CGI script**. (ScanSoft OmniForm 4.01 Sample Perl Scripts, page 1).

Not only does OmniForm fail to generate program code external to the form, but the archived website further provides evidence of the difficulty of creating the external program code. The website stated that "CGI applications are powerful and useful, but are not intended for novice computer users. **They require a high level of technical knowledge to be implemented properly**." (ScanSoft OmniForm 4.01 Sample Perl Scripts, page 1) (emphasis added). ScanSoft provided sample scripts "as is" due to the difficulty of implementation.

Additionally, it is respectfully submitted that Larsen also fails to disclose generating program code. Strong describes a CGI program that is external to a form to perform validation of a submission of the form. Strong, however, does not describe any other actions beyond validation – this is left to "Handlers" which are simply pointers to separate plug-in modules that will perform other actions beyond the validation (Col 10 line 66 to Col 11 line 5). Strong fails to disclose **automatically generating program code** to carry out the actions associated with the identified input fields wherein the actions are one or more from a group of two or more. Strong describes a wizard that generates the registry keys for storing the validation information, but does not teach automatic code generation for performing any other actions. (*see* Strong, col. 8, line 5).

Even if the validation is considered one type of action for which program code is automatically generated, there are no other types of actions described. There is no description of a user interface being presented to allow identification of any other types of action, or for automated generation of a program code that will be responsive to a subsequent submission of the form. Strong, similar to the OmniForm web pages cited above, states that CGI programs are complex to develop and require knowledge of a programming language. (Strong, col. 1, lines 38-46). Strong's reference to a CGI program being external to the form does not make obvious automatically generating, independently of the form authoring tool, program code that is external to the form.

Accordingly, Applicant submits that both the OmniForm website and Strong provide evidence not only that the references lack this claim feature, but that the discussed claim feature is not obvious in light of the level of difficulty in implementing a CGI script that will work with an arbitrary form created independently. Therefore, none of the references, alone or combined, discloses automatically generating, independently of the form authoring tool, program code to carry out the one or more actions, wherein the program code is external to the form and independent of the form authoring tool.

Accordingly, Applicant respectfully submits that the rejection of claim 126, 132, and 137 has been overcome.

Given that claims 127-129, 131, 133-136, 138-140, 142, and 144-154 are dependent upon claims 126, 132, and 137, and include additional features, Applicant respectfully submits that the rejection of claims 127-129, 131, 133-136, 138-140, 142, and 144-154 has been overcome for at least the same reasons as above.

Furthermore, Applicant disagrees with the specific rejection of dependent claims. In the interest of brevity, Applicant will only provide arguments for a few of the dependent claims and reserve the right to provide arguments for the remainder of the dependent claims at a later time should the rejections be maintained.

Regarding claims 131, 136, and 142, the combination of OmniForm and Larsen fails to disclose a method, server, or machine readable medium to determine that the generated program code is consistent with the form and generating an alert if the generated program code is not consistent with the form. The Office Action alleges that OmniForm teaches validation options for validating input and that if the input does not validate, the user is notified. (Office Action dated 9/18/08, page 5).

Applicant respectfully submits that OmniForm describes a validation option for determining that **a user input into a text field complies with a rule** – e.g., a mandatory field may not be left blank, the type or size of input, etc. Nevertheless, OmniForm does not disclose generating an alert when it is determined that **the generated program code is not consistent with the form**. As discussed above, OmniForm does not generate program code external to the form. Additionally, OmniForm does not describe determining consistency between a form and generated program code. While the CGI Wizard may use a form to fill in predetermined variables in a template CGI script, it does not determine consistency or generate an alert if an inconsistency is found.

Accordingly, Applicant respectfully submits that the rejection of claim 131, 136, and 142 has been overcome for these reasons in addition to the arguments above relating to the independent claims.

Claims 128, 135, and 139 stand rejected under 35 U.S.C. 103(a) as being unpatentable over OmniForm, in view of Larsen, in further view of Strong, in further view of PR NewsWire article (hereinafter "PR NewsWire), March 22, 1999, ProQuest Direct, pages 1-5.

Given that claims 128, 135, and 139 are dependent upon claims 126, 132, and 137, and include additional features, and given that PR NewsWire fails to remedy the shortcomings of OmniForm, Larsen, and Strong discussed above, Applicant respectfully submits that the rejection of claims 128, 135, and 139 has been overcome for at least the same reasons as set forth above.

Claim 147 stands rejected under 35 U.S.C. 103(a) as being unpatentable over OmniForm, in view of Larsen, in further view of Strong, in further view of Davis, U.S. Patent No. 5,796,952 (hereinafter "Davis").

Given that claim 147 is dependent upon claim 126, and includes additional features, and given that Davis fails to remedy the shortcomings of OmniForm, Larsen, and Strong discussed above, Applicant respectfully submits that the rejection of claim 147 has been overcome for at least the same reasons as set forth above.

CONCLUSION

Applicant respectfully submits that all rejections have been overcome and requests reconsideration of the present application.

If there are any additional charges, please charge them to our Deposit Account Number 02-2666. If a telephone conference would facilitate the prosecution of this application, the Examiner is invited to contact Ryan W. Elliott at (408) 720-8300.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: July 1, 2009

/Ryan W. Elliott/
Ryan W. Elliott
Reg. No.: 60,156

1279 Oakmead Parkway
Sunnyvale, CA 94085
(408) 720-8300